MIFARE® PLUS DEMO MANUAL

Version 1.0

GUI examples:

C#:

https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-mfp-examples-csharp-gui.git

C++:

https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-mfp-examples-cpp-gui.git

Console examples:

C:

https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-mfp-examples-c.git

C#:

https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-mfp-examples-csharp-console.git

# Table of contents

Digital Logic Ltd.

About Mifare® Plus cards

- Keys can be stored as MIFARE® CRYPTO1 keys (6 bytes per sector) and AES keys (16 bytes per sector) - Memory structure identical to MIFARE® Classic 4K (sectors, blocks)
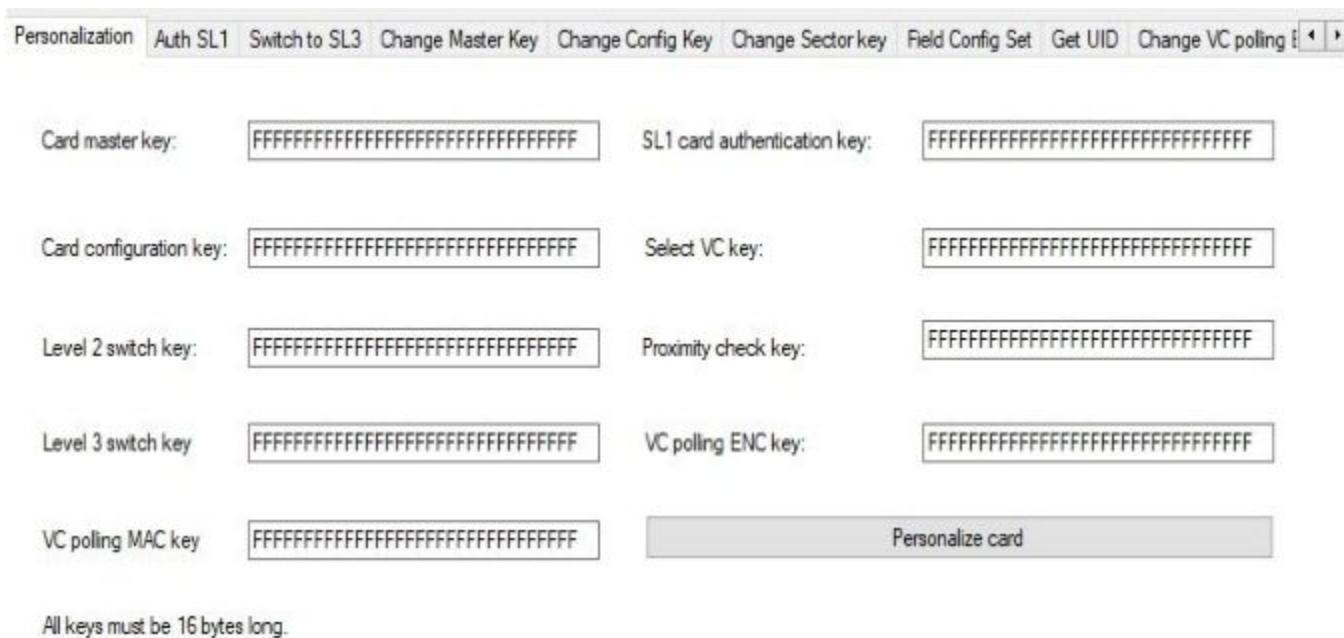
Application demo

When you simply double click on the .exe and the program starts, you shall see that we have option of Reader Open the usual way, with our ReaderOpen() function, or you can check the "Use Advanced options" and use our Advanced options parameters to open communication port with ReaderOpenEx function, provided you input correct parameters in Advanced options fields (reader type, port name, port interface and additional argument). When the port has been successfully open, it will print out "ReaderOpen/ReaderOpenEx was successful" in lower left corner and reader description in the text box on the bottom of the application. After successful port open, it will start continuous search for card in its NFC field. If found - it will print out cards type and UID in upper left corner. If not, it will show UFR_NO_CARD in the same place where cards type would be. For example:
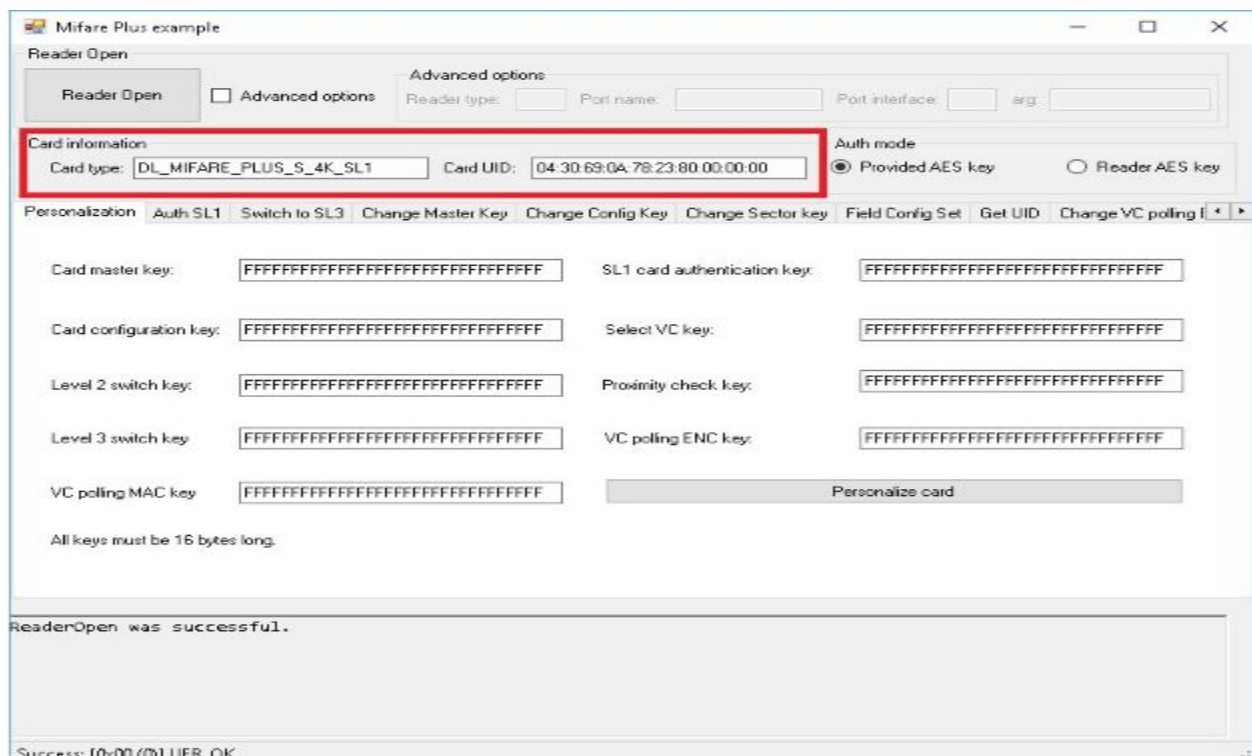
From the above picture, we can see all of the tabs dedicated to working with MIFARE® Plus cards. Keep in mind that not all functions mentioned in this demo work with all three cards security levels. Also, all functions on these tabs, except for the last three - "Data read, data write, Write keys into reader" depend on Authentication mode that can be chosen in upper right corner. If we check "Provided AES key" then we will use functions with provided key required. And vice versa, if we check "Reader AES key" then our functions will use reader key that we select on our tabs.
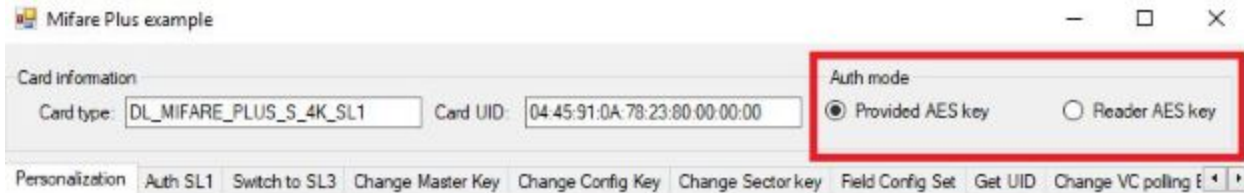
Let's start from Personalization tab:

On this tab we have 9 keys that the card uses for its intended use and purpose. Inputting all of them and clicking the "Personalize card" button will have our card set up for future use. All of these keys have their use explained in their names. For example "Card configuration key" is used for changing cards configuration settings, such as Field configuration settings, authorization of changing VC polling ENC/MAC keys, reader AES keys, etc.. Cards default security level is Security Level 0. Cards security level automatically updates to level 1 as soon as we complete card Personalization. On this level we can also choose AES authentication, which is by default used on SL3. We accomplish that on the next tab.

On this tab, depending on what authentication mode we selected in the upper right corner, we use one of these keys to enable MIFARE® Plus authentication with AES keys instead of CRYPTO1 while it's in SL1 mode. MIFARE® Plus supports all MIFARE® Classic value-block operations either in SL1 or SL3 mode. After security upgrade to SL3 mode, MIFARE® Plus uses Advanced Encryption Standard (AES) for authentication, data integrity and encryption. In our demo we switch to SL3 on the next tab - "Switch to SL3".

Usage of this tab is same for the previous. We check the authentication mode in the upper right corner, enter or select key that we will use to authenticate this process and click on the button.

On next tab, we can change cards Master key. To do so, we must enter previous, old master key so we can authorize usage of the new key that we enter.

The same principle goes for all of the key changes mentioned in our demo with an exception of changing sector key - where we have to specify for which sector we shall be changing AES key.

As for the Field configuration settings, in other words use choice of Random UID that Desfire cards have or regular UID, we authorize that setting with our cards Configuration key and select one of our options on top: Use random UID or standard UID:

The random UID feature means that the cards UID is new every time it is powered on, which means every time it is used. To randomise the UID means that everytime the card is used, a new UID is recorded. This each activity of the card cannot be linked to the user, and he chain of activity cannot be traced to the card. Card needs to be in SL3 mode to use this feature. Only way to get cards true UID with this feature enabled is to get UID with usage of VC polling ENC & MAC keys. We will demonstrate this in next tab "Get UID".

For example, our card generates a random UID like this:

#1
Mifare Plus example

Card information
Card type: DL_MIFARE_PLUS_S_4K_SL3   Card UID: 08:9C:C7:37:00:00:00:00:00:00

#2
Mifare Plus example

Card information
Card type: DL_MIFARE_PLUS_S_4K_SL3   Card UID: 08:9C:C7:37:00:00:00:00:00:00

Each time our card gets into NFC field and gets powered on - it shall get new randomized UID. To see cards true/factory UID, we enter ENC and MAC key or select them by their index if they are stored in the reader.
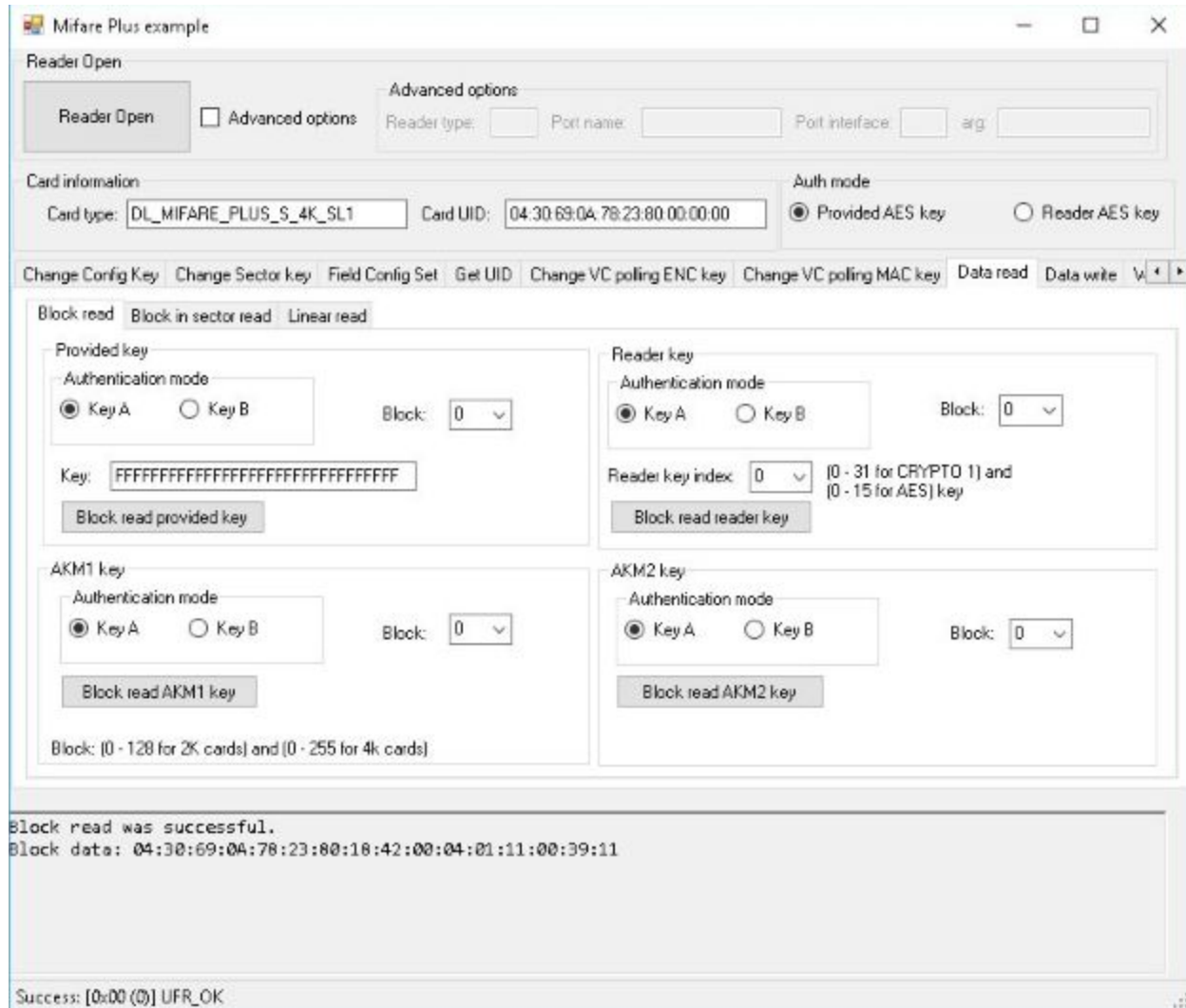
From the image above we can see that UID has changed and will be so until we enable random UID again. From the text box below we can see that our settings were set successfully.

Now we shall demonstrate Data reading and writing. This section has an exception of different authentication modes and different reading/writing modes influenced by our choice of previously mentioned authentication modes.

Here we have Key A and Key B authentication modes, as for the modes of reading/writing we have Provided key mode, Reader key mode, AKM1 mode and AKM2 mode. For reference of these settings such as KeyA/B authentications and AKM1/2 mode, you can find explanations in our UFR Series NFC reader API here: https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-doc
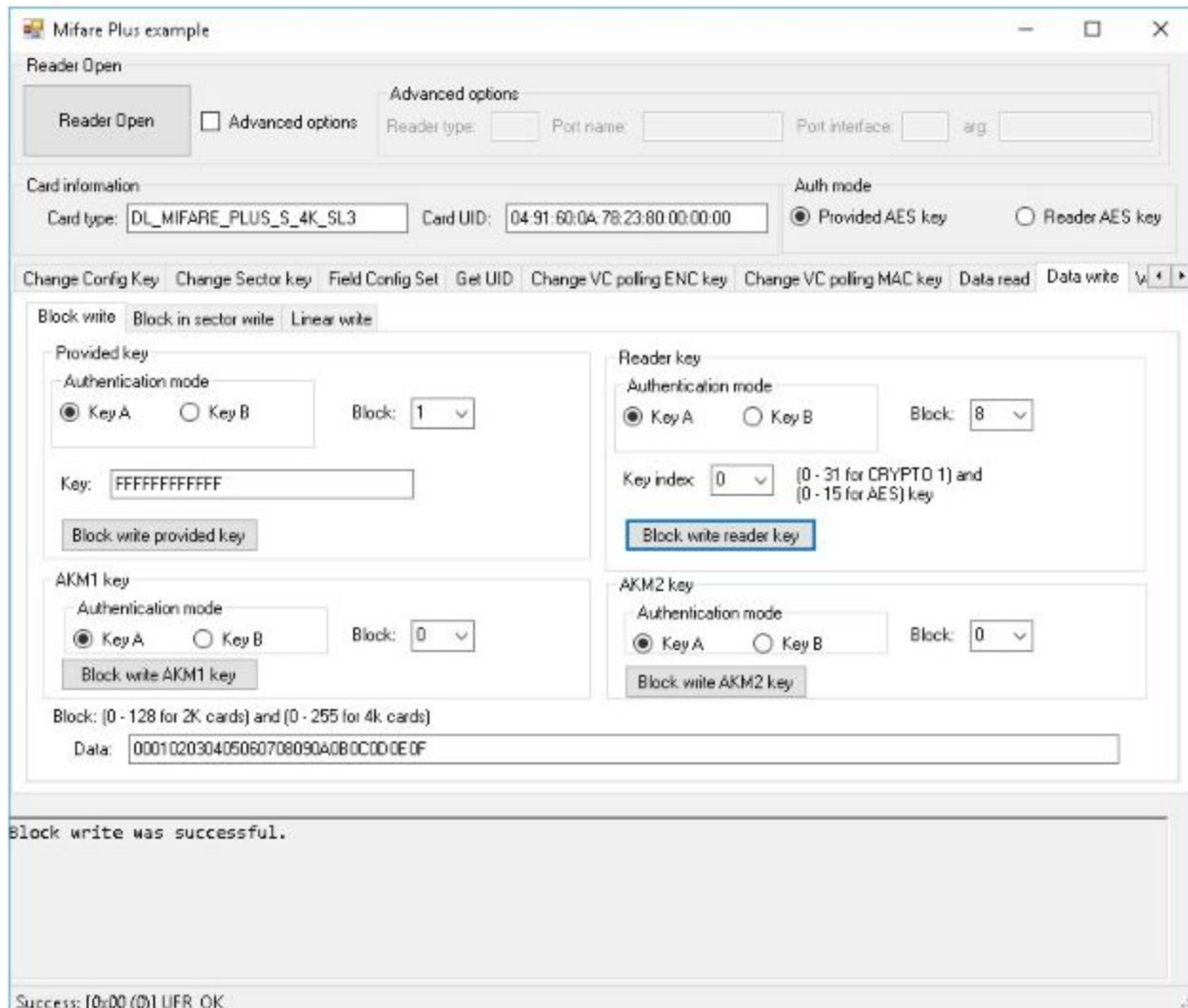
Example for Block read:

Here we see that we successfully read block 0 with one of these functions, and block data has been written in the bottom of the screen.

Similar principle goes for Data writing with exception of entering data we wish to write in our card, for example Block write:

We just choose block that we wish to write into, enter or select key if needed and click on the provided button. If we go back to reading, we will see our data written in,for example, block 1 successfully.

For storing keys in our reader, we have dedicated tab "Write keys into reader" where we can enter either CRYPTO1 or AES keys, and lock further key input in our device or unlock it.

As a reminder: CRYPTO1 key is 6 bytes long, AES key is 16 bytes long, and our password input is 8 characters, for the sake of simplicity.

API reference for functions used in this demo: https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-doc

Revision history

Date	Version	Comment
2019-04-09	1.0	Base document

Digital Logic Ltd.